Final Report

Author: Jaskirat Kaur, Grey Xu, Qilin Zhou

Abstract

Climate change poses significant threats to our marine ecosystems, particularly to coral reefs, many of which are experiencing bleaching. This project aims to address this issue using deep learning techniques. The project is divided into two parts: Classification of coral health status and Generative Models to visualize potential unhealthy states of currently healthy corals.

In the classification part, we focus on identifying healthy and unhealthy corals and further classifying coral species. For the healthy-unhealthy coral classification, we used four models—Custom, YOLO8, EfficientNet, and Vision Transformer—to compare their performance. The Vision Transformer model achieved the best performance, with a loss of 0.32 and a test accuracy of 85.87% on a dataset of 923 images. For species classification, we applied YOLO8 with the Coral Net dataset to enable our user interface to display species types on the web application.

The second part involves Generative Models. We developed custom GANs to visualize how unhealthy coral reefs would look if they were healthy and vice versa. Additionally, we utilized pre-trained Cartoon GANs, named after directors like Hayao Miyazaki, Mamoru Hosoda, Satoshi Kon, and Makoto Shinkai, to transform images into specific artistic styles. These models offer a unique way to explore ocean species in digital media.

We also conducted additional analysis on the detection model beyond the main objectives. Due to limited computational resources and time, we concentrated on seven species from the FathomNet dataset, which includes labeled bounding boxes. Despite our efforts, the custom model faced challenges in achieving high accuracy, so we directly applied YOLO8 to demonstrate object detection on the web app.

To integrate these efforts, we built an interactive Streamlit app. Users can upload images, and videos, or choose from YouTube to perform classification, apply GAN transformations, or detect species bounding boxes using different models on various datasets.

Grey focused on EDA and model selection for the healthy coral datasets by comparing models and managing the ML flow pipeline for classification. Jaskirat worked on EDA and detection problems with the FathomNet dataset, fine-tuning the Vision Transformer, and generating

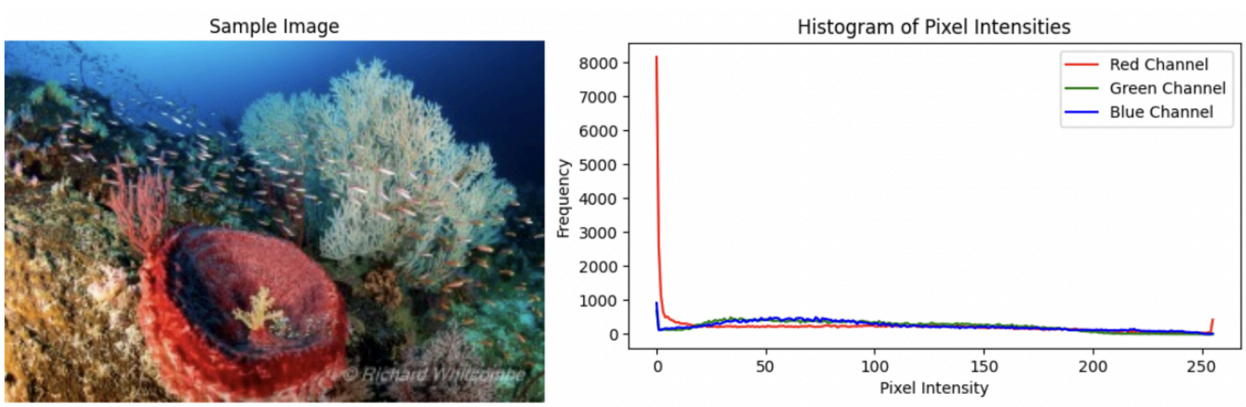
evaluation metrics. Qilin customized and deployed the GAN model and built the web application for the project. Collaboratively, we worked on the slides and the report.

1. Exploratory Data Analysis

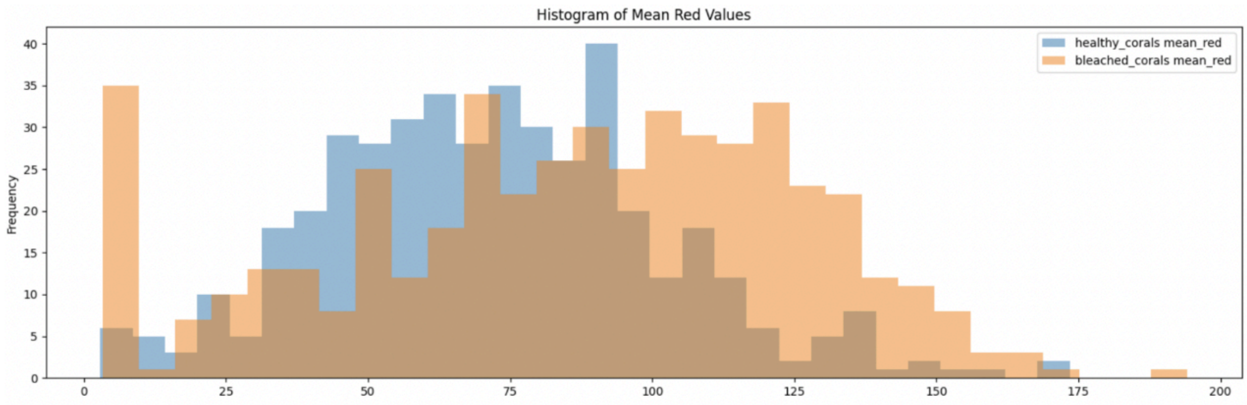
Section 1.1 Coal health

To classify coral's health status as well as generate hypothetical images, we used a dataset consisting of 923 images, categorized into two groups: 438 images of healthy corals and 485 images of bleached corals. All images are in JPEG format and have a maximum dimension of 300 pixels in either width or height. These images were sourced using the Flickr API, which is available for non-commercial use by outside developers. This is also the main dataset we used for our classification and generation.

For the exploratory data analysis (EDA) of the coral health dataset, we performed several key tasks to understand the characteristics of healthy and unhealthy corals. First, we examined a random sample from each category, analyzing image properties such as format, size, and color distribution. We visualized sample images and plotted their color histograms to identify differences in color channels as shown below.



Next, we conducted a comprehensive analysis of all images in the dataset, focusing on mean and standard deviation values for the red, green, and blue color channels. We found that healthy corals exhibited a broader distribution of mean red values, indicating higher variability in red tones, while bleached corals showed a skew towards lower red values, a typical sign of bleaching as shown in the below histogram.



To further understand these differences, we performed logistic regression to assess the impact of color channel statistics on coral health. The regression analysis revealed that an increase in mean red value significantly increased the likelihood of bleaching. In contrast, the mean green and blue values did not significantly affect the likelihood of bleaching. However, higher variability in the red and blue channels was associated with a lower likelihood of bleaching, whereas greater variability in the green channel increased the likelihood.

Logit Regression Results

Dep. Variable:	label_encoded	No. Observations:	923
Model:	Logit	Df Residuals:	914
Method:	MLE	Df Model:	8
Date:	Sat, 27 Apr 2024	Pseudo R-squ.:	0.2050
Time:	21:41:02	Log-Likelihood:	-507.66
converged:	True	LL-Null:	-638.58
Covariance Type:	nonrobust	LLR p-value:	5.332e-52

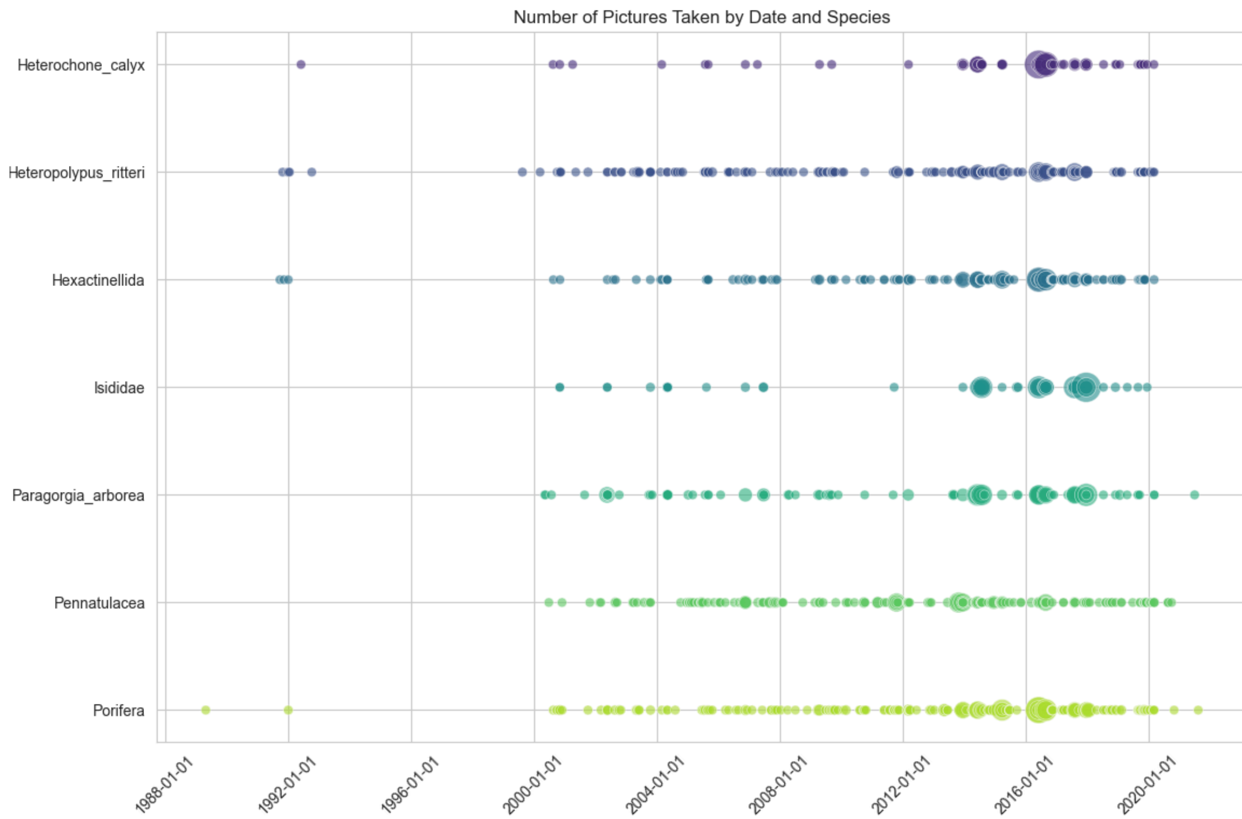
	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.1204	0.621	-0.194	0.846	-1.337	1.097
mean_red	0.0167	0.003	5.495	0.000	0.011	0.023
mean_green	-0.0050	0.006	-0.871	0.384	-0.016	0.006
mean_blue	0.0033	0.005	0.691	0.490	-0.006	0.013
std_red	-0.0536	0.007	-7.796	0.000	-0.067	-0.040
std_green	0.1175	0.013	8.780	0.000	0.091	0.144
std_blue	-0.0758	0.010	-7.379	0.000	-0.096	-0.056
width	0.0061	0.001	4.242	0.000	0.003	0.009
height	-0.0084	0.002	-4.508	0.000	-0.012	-0.005

Section 1.2 Fathom net

This dataset is primarily used for object detection. The FathomNet dataset consists of images collected in the Monterey Bay Area from the surface to a depth of 1300 meters. The images include bounding box annotations for 290 categories of bottom-dwelling animals. The training data, collected from 0-800 meters, and the evaluation data, spanning 0-1300 meters, reflect overlapping but distinct species distributions due to depth-related habitat variations. Test images are similarly sourced from the entire 0-1300 meter range. Due to limited computational resources and time, we focused on seven species from the FathomNet dataset, which includes labeled bounding boxes.

Through the FathomNet API, we extracted a total of 24,374 images for the selected seven species: Pennatulacea (Sea Pens), Porifera (Sponges), Hexactinellida (Venus Flower Basket), Paragorgia arborea (Bubblegum Coral), Isididae (Bamboo Coral), Heterochone calyx (Fingered Goblet Sponge), and Heteropolypus ritteri (Ritter's Soft Coral). These images are frames taken from videos. We also pulled their bounding box and geographic data to perform temporal and geospatial analysis along with performing detection. The distribution of images among these species was relatively balanced, which is crucial for training robust detection models. The dataset was divided into 17,061 training images, 4,874 validation images, and 2,439 test images. This structured division ensures comprehensive phases for training, validation, and testing, allowing for effective development and evaluation of our detection models.

The visualizations below provide a detailed overview of the dataset. The scatter plot illustrates the temporal distribution of images for each species. We notice that the highest number of images were taken around 2016, whereas the period before 2000 saw very few images. The horizontal bar plot displays the image count for each species, confirming the balanced nature of the dataset. These visualizations ensure a balanced representation across species, facilitating the training and evaluation processes. The dataset's extensive range and balanced distribution make it a valuable asset for developing precise object detection models.


Number of Pictures Taken by Date and Species
Heterochone_calyx
Heteropolypus_ritteri
Hexactinellida
Isididae
Paragorgia_arborea
Pennatulacea
Porifera
1988-01-01
1992-01-01
1996-01-01
2000-01-01
2004-01-01
2008-01-01
2012-01-01
2016-01-01
2020-01-01

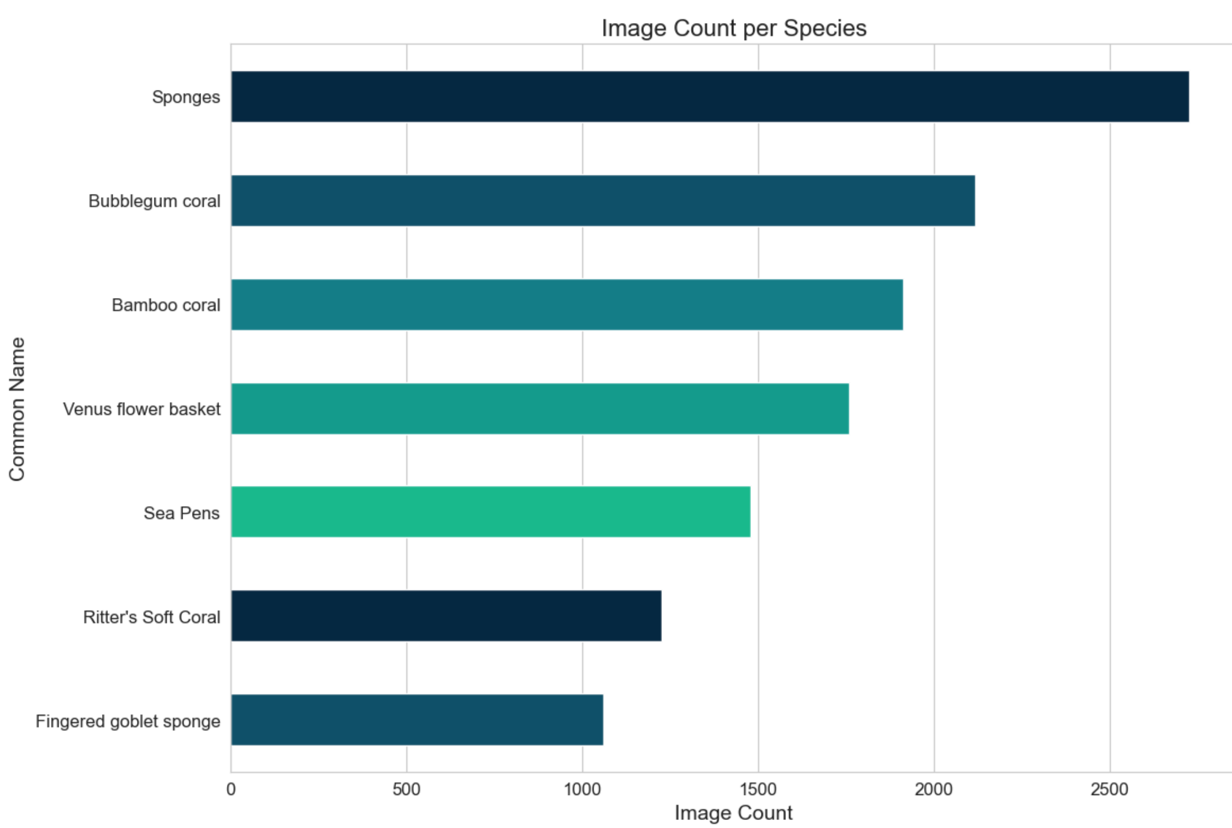


Image Count per Species
Common Name
Sponges
Bubblegum coral
Bamboo coral
Venus flower basket
Sea Pens
Ritter's Soft Coral
Fingered goblet sponge
0
500
1000
1500
2000
2500
Image Count

Section 1.3. Coral net

We further employ coral net dataset to classify the species type in our images. The CoralNet dataset contains over 300,000 images of coral reefs used for species classification. These images are sourced from extensive benthic surveys and are annotated and labeled to facilitate automated analysis and monitoring of coral reef ecosystems. The dataset is accessible through the CoralNet website developed by UCSD. We used a subsample of those (around 8000 images) to classify the common corals and visualize the species name in the web application.

Section 1.4 General Marine

In order to allow our demo to showcase various marine objects, we applied the general marine dataset, which aims for precise object localization in diverse underwater scenes. This dataset includes seven distinct classes: Fish, Crab, Human, Trash, Jellyfish, Coral Reef, and Fish Group. It is split into training (70%), validation (10%), and test (20%) sets. The images have been pre-processed with auto-orientation, resized to 640x640, and contrast-adjusted using contrast stretching, with 798 images in total. Sourced from the Brackish Dataset by Aalborg University and the Trash Debris Dataset from the University of Minnesota.

2. Model Training and Evaluation

Section 2.1. Classification

To classify the health status of corals, we employed several models to ensure a comprehensive evaluation and comparison. We used a custom CNN model specifically constructed for this task, which features multiple convolutional layers, batch normalization, dropout, and max pooling to effectively extract and process features from the coral images. Alongside this, we utilized the YOLO8 model, known for its real-time object detection capabilities, which leverages the Ultralytics package to enhance classification accuracy. EfficientNet, a family of convolutional neural networks, was also employed using pre-trained models (b0, b1, n2) with applied dropout rates to prevent overfitting while maintaining high performance. Lastly, we incorporated the Vision Transformer model, a state-of-the-art approach that utilizes self-attention mechanisms to handle image classification tasks with high precision and efficiency. Each of these models was chosen for its unique strengths, providing us with a broad spectrum of performance metrics to identify the most effective model for coral health classification.

The custom model architecture comprises multiple layers designed to extract and process features from the input images. The input layer accepts images of shape 128x128x3, followed by three convolutional layers with 32 neurons each, using ReLU activation and L1/L2 regularization. Each convolutional layer is followed by batch normalization and a 2x2 max pooling layer to reduce dimensionality. Dropout layers with a 20% rate are added after each pooling layer to prevent overfitting. The final convolutional output is flattened into a 1D array and passed through a dense layer with a single neuron using sigmoid activation for binary classification.

The preprocessing steps for ViT involved resizing, image augmentation, as well as resplitting the data into 70% training data, 10% validation and 20% test data.

The results of the models are summarized in the table below. The custom CNN model and EfficientNet exhibited comparatively lower accuracy rates of 61.96% and 47.28%, respectively. YOLO8 demonstrated relatively strong performance with a test accuracy of 78.26% and a lower test loss of 0.53. The Vision Transformer model emerged as the best-performing model, achieving a test loss of 0.32 and a test accuracy of 85.86%. Additionally, this model achieved an F1 score of 0.8421, with a confusion matrix showing 42 correctly classified bleached corals and 28 healthy corals, but 14 misclassified bleached and 8 healthy corals. The Vision Transformer demonstrated a strong capability in classification, with an AUC of 0.8537. The advantages of the Vision Transformer include its ability to capture long-range dependencies in images, making it particularly effective for identifying complex patterns of coral healthiness.

	Custom	Efficient Net	Yolo8	Vision Transformer
Test loss	0.77	0.83	0.53	0.32
Test Accuracy	61.96	47.28	78.26	85.86
Test Precision	0.79	1.0	0.89	0.84
Test Recall	0.26	0.47	0.72	0.82

Table: Model metric comparison on classifying coral health status

However, it requires substantial computational resources and large datasets for optimal performance. Given that the ViT has proven to work well with extremely large datasets, our model's accuracy could be improved further with more data and computational resources. Early stopping was applied at epoch 6 to prevent overfitting, resulting in a final validation loss of 0.3219, precision of 0.8421, and recall of 0.8205. These results, supported by visualizations of the confusion matrix and ROC curve, highlight the ViT model's effectiveness in accurately classifying coral health, making it a promising tool for marine conservation efforts.

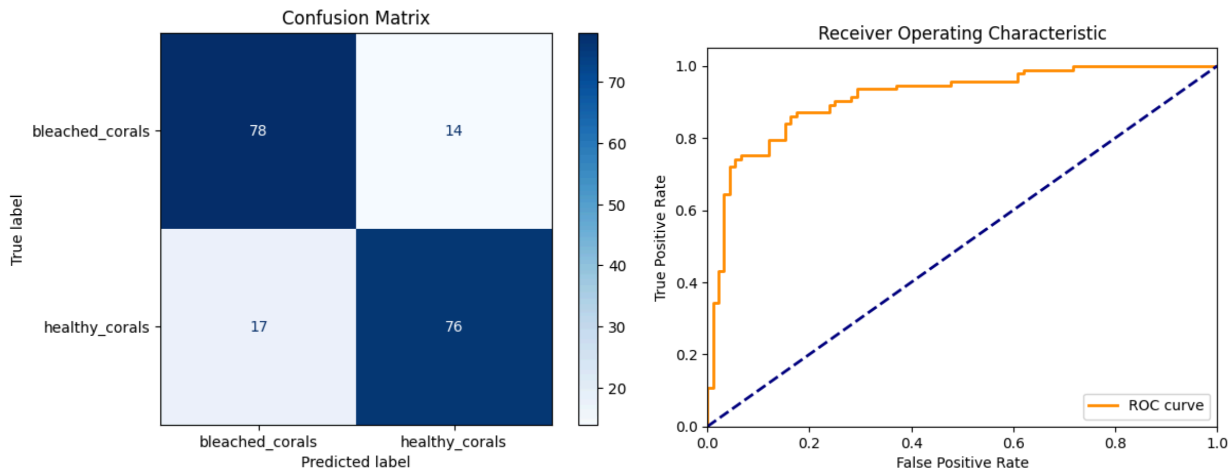


Figure: Confusion Matrix and ROC Curve for ViT

Section 2.2 Generation

In the training process of our custom Coral CycleGAN, we focus on transforming images between healthy and bleached coral states. Initially, both generators create synthetic images: the healthy generator synthesizes healthy coral images from bleached inputs, and the bleached generator does the opposite.

The architecture of the generator designed for this task includes an initial layer that takes 256x256 pixel images with three color channels. The down-sampling section consists of several layers that progressively increase the number of filters from 64 to 512 while reducing the image resolution. This begins with a 64-filter layer with a 4x4 kernel and a stride of 2, applying LeakyReLU activation without instance normalization. Subsequent layers continue with the same kernel size and stride but increase the filter count, incorporating instance normalization and LeakyReLU activation. Additional layers, each equipped with 512 filters, further compress the features to capture more complex patterns.

The up-sampling section of the generator works to reconstruct the image to its original size, enhancing details through layers that gradually decrease the number of filters from 512 to 64. These layers employ a 4x4 kernel, a stride of 2, instance normalization, and ReLU activation to restore spatial dimensions and refine the output. The final output is achieved through a Conv2DTranspose layer with 3 filters, a 4x4 kernel, and a stride of 2, using a tanh activation to normalize pixel values between -1 and 1.

The discriminator starts with an input layer for 256x256 pixel images and uses a series of down-sampling layers that progressively increase filter size and apply instance normalization and LeakyReLU activation to enhance its ability to classify images accurately. Additional layers with zero padding and a final convolutional layer focus on detailed features, culminating in a 30x30 feature map that assesses different image patches, determining their authenticity.

Over the course of 25 epochs, our Coral CycleGAN's training logs reveal a consistent improvement in performance, with generator losses steadily decreasing from initial values around 4.72 and 4.62 to approximately 1.99 by the final epoch. The discriminator loss metrics stabilizes in the mid-0.64 range towards the later epochs. This pattern suggests effective learning and model stabilization without significant signs of overfitting. The figures below showcased a significant and well-executed transformation.

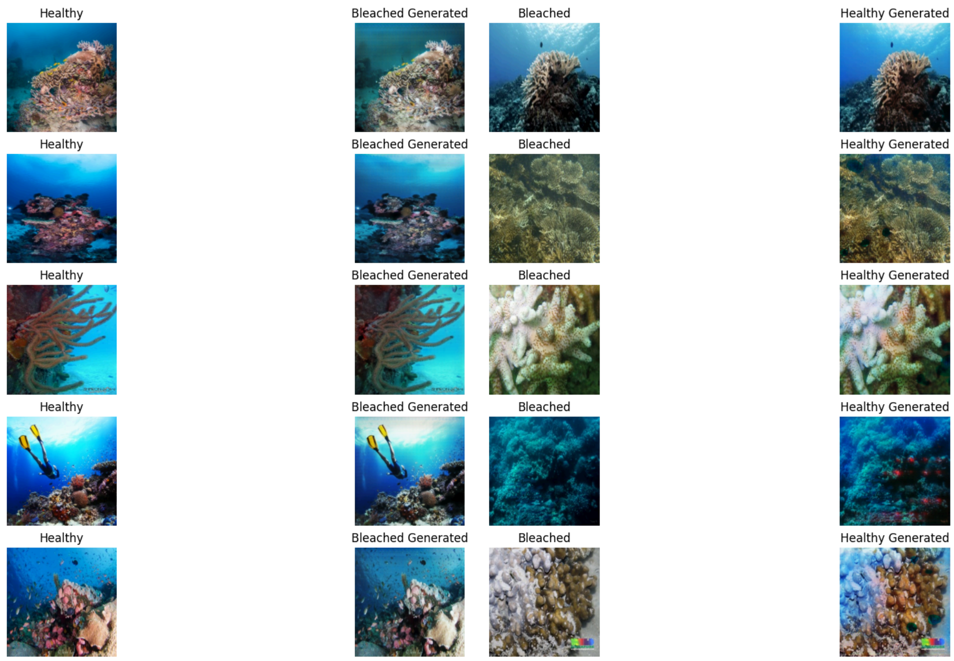


Figure: Example of generated images using Custom CycleGAN

Previous researchers have similarly employed CycleGAN for data augmentation in disease diagnosis. In 2020, Quan Huu Cap and colleagues developed LeafGAN for practical plant disease diagnosis, incorporating a specialized attention mechanism (Cap et al., 2020). This mechanism allowed the GAN to concentrate transformations solely on pertinent areas of plant leaves, preserving the background intact. Currently, our Coral CycleGAN does not utilize such attention mechanisms. However, incorporating this feature could be advantageous, as our videos contain multiple species that do not require diagnosis, allowing for more targeted and efficient processing.

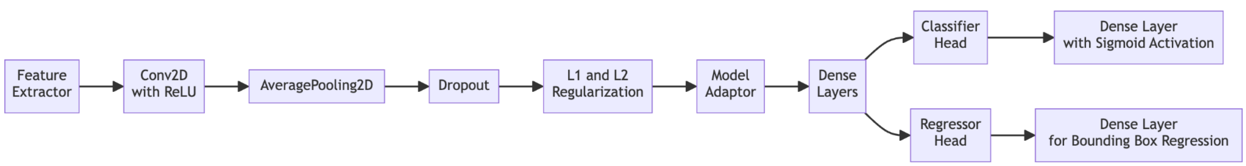
Section 2.3 Detection

2.3.1 Custom Model for Species Detection

Model Architecture

We developed a custom model designed to identify and classify coral species in underwater images. The architecture of our model involves several components. The feature extractor consists of a series of convolutional layers to extract spatial features from the images. These layers include Conv2D layers with ReLU activation, AveragePooling2D layers for downsampling, Dropout layers to prevent overfitting, and L1 and L2 regularization to penalize large weights. Following this, the model adaptor, comprised of dense layers, processes the extracted features further, preparing them for the final output layers. The classifier head uses a dense layer with sigmoid activation for binary classification, while the regressor head utilizes a dense layer for bounding box regression, predicting the coordinates of the bounding boxes. The model was trained with an input size of 244x244 pixels using the FathomNet dataset, which contains labeled bounding boxes for various marine species.

The model was trained with an input size of 244x244 pixels using the FathomNet dataset, which includes labeled bounding boxes for various marine species.



Training and Evaluation

Training the custom detection model for 10 epochs took approximately one day for only 300 images due to the model's complexity and the computational load. The Intersection over Union (IoU) scores, which measure the overlap between predicted and actual bounding boxes, were extremely low. Most IoU scores were 0.0000, indicating poor detection performance. This poor performance is largely due to the size and nature of the training data. The FathomNet dataset images, extracted from videos, often contain multiple objects besides coral, leading to obscured views and cluttered scenes, causing the model to misdetect.

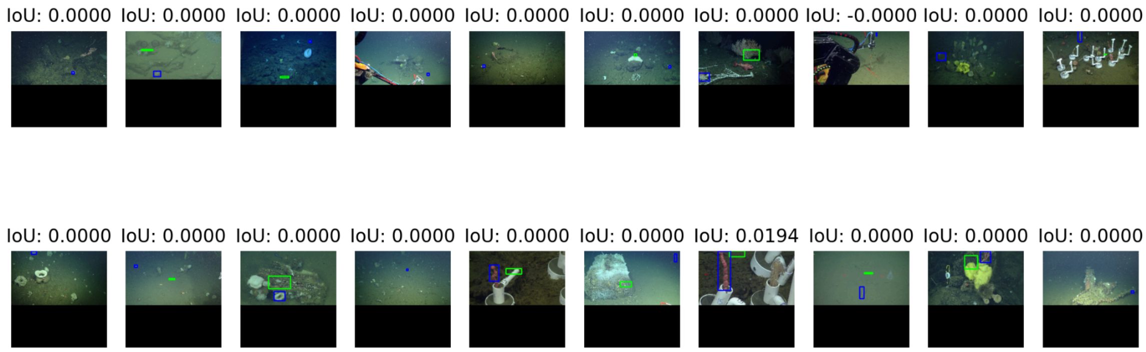


Figure: Poor IoU results

Evaluation Results
The evaluation on the test set revealed significant shortcomings, with most IoU scores being 0.0000, demonstrating poor overlap between predicted and actual bounding boxes. The predicted bounding boxes (in blue) rarely matched the ground truth boxes (in green), highlighting frequent misdetections due to the complex nature of the training images with multiple non-coral objects in the background.

Future Work
Future enhancements will focus on optimizing the model architecture and reducing training time. Improvements could include exploring more efficient architectures, increasing training data to the full size of the dataset(~24,000 images for 7 species), implementing advanced data augmentation techniques to improve robustness, and leveraging pre-trained models on similar datasets to enhance detection accuracy.

3. Model Operations

Section 3.1 Model Management

To systematically manage our models and hyperparameter tuning, we employed MLFlow. This tool enabled us to record each trial and visualize the results, facilitating effective tracking of experiments. Parameters such as batch size, number of epochs, learning rate, regularization, and dropout rates were meticulously adjusted to address overfitting and optimize model performance. MLFlow also allowed us to version our models, ensuring that any updates or changes were documented and could be reproduced, thereby maintaining a robust and transparent model development process.

The accompanying image shows an example of our model management process using MLFlow, where we visualized the relationship between different hyperparameters and their impact on test loss and test accuracy. The parallel coordinates plot highlights how variations in parameters like batch size, dropout rate, and learning rate influence the model's performance, allowing us to identify the optimal settings for the highest accuracy.

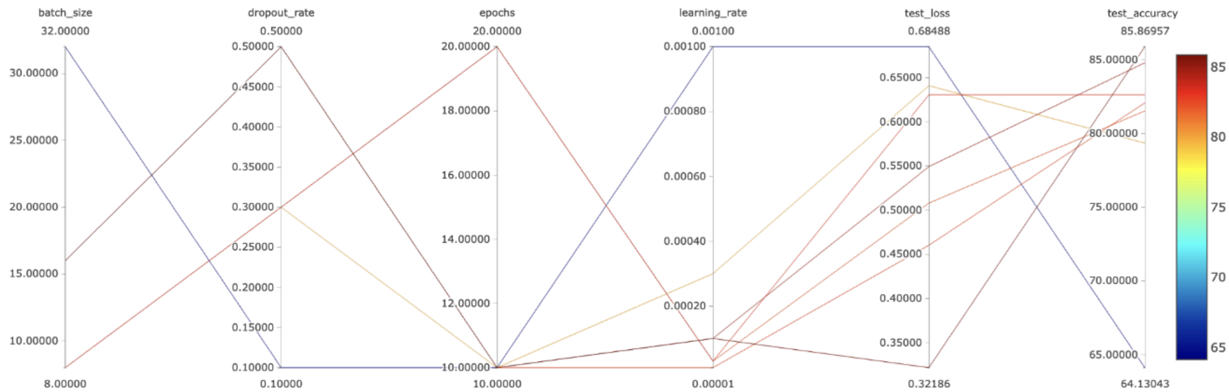


Figure: Model Parameter Tuning using MLflow

Section 3.2 Model Deployment

For model deployment, we leveraged Google Colab to harness its powerful computational resources, especially for training larger models like GANs. Google Colab provided an accessible and scalable platform for handling complex computations and large datasets, enabling efficient model training and experimentation. The system was configured to automatically save the highest-performing model to Google Drive. Additionally, there are opportunities to deploy on cloud platforms such as AWS or Google Cloud, which could further scale the solution and enhance its accessibility.

For visualization, we used Streamlit to create an interactive and user-friendly interface. This proof-of-concept deployment allowed users to upload images or videos, perform classification, apply GAN transformations, and detect species bounding boxes. Streamlit provided a seamless integration with our models, making it easy to upload images or videos to demonstrate model performance and ensure accessibility for end-users. Moreover, we utilized Streamlit Cloud to deploy the web application with a public url, making it accessible to a broader audience, which will be explained below.

Section 3.3 User interface

3.3.1 Web Application Outline

For the Streamit application, we have developed three interactive pages. The first page focuses on computer vision models, where users can engage with four different datasets for specific tasks. In the section for healthy and bleached coral datasets, users have the option to select from three pre-trained models—ViT, EfficientNet, and YOLOv8,—as well as one custom model. For style transfer tasks, users can utilize our custom Coral CycleGAN to generate images depicting coral pollution recovery or opt for pre-trained Cartoon GANs for stylized coral imagery. The remaining tabs are dedicated to detection: YOLOv8 is available for detecting species in FathomNet and CoralNet, and an additional tab offers OWL-ViT for zero-shot detection, emphasizing that detection is not the primary focus of our project. This interface supports inputs in the form of images, videos, and YouTube videos.

The second page is dedicated to coral health status, providing a visual and statistical analysis of pixel intensities between bleached and healthy corals using logistic regression. And the final page, the species map, allows users to interact with the FathomNet API to explore the most frequently camera-captured species near Monterey Bay at various depths, displayed on a map. This setup provides a comprehensive toolset for both casual users interested in marine biology and researchers conducting in-depth studies.

3.3.2 Application Use Instruction

We have made our code and documentation publicly available in our [GitHub repository](). Users can git clone and run the app file on their local machines, which is the best practice for now due to storage limitations to run in the cloud. We have made a public URL on Streamlit Cloud, but it may need frequent reboost under the 1GB constraint.

Another option would be to run in our shared drive, which is the fastest approach. For users with access to our Google Drive, they could run *app.ipynb* under the application directory of “Marine Species Detection Open CV” folder.

4. Conclusion

We leveraged deep learning techniques to address the pressing issue of coral bleaching due to climate change. By classifying coral health status and developing generative models, we aimed to provide valuable tools for marine conservation efforts. Our analysis utilized several advanced models, including Custom CNN, YOLO8, EfficientNet, and Vision Transformer, with the Vision Transformer emerging as the best-performing model for classifying coral health. Additionally, we developed custom GANs to visualize potential unhealthy states of currently healthy corals, offering unique insights into the visual impact of bleaching.

One of the significant challenges we faced was the large image size, which demanded substantial computational power. Leveraging cloud computing resources like Google Colab was essential for efficient training and experimentation. Despite our efforts, achieving good Intersection over Union (IoU) results with the custom detection model proved difficult. In contrast, YOLO8 demonstrated reliable performance without such issues, highlighting its robustness in object detection tasks.

Additionally, the resolution of the healthy coral images, capped at 300x300 pixels, posed a challenge for improving classification accuracy. The relatively low resolution limits the amount of detail available for the models to learn from, which can hinder their ability to distinguish subtle differences between healthy and bleached corals. To mitigate this, future work could explore techniques such as super-resolution to enhance image quality before classification or collect higher-resolution images to provide more detailed training data.

Further qualitative evaluation of the GAN model is necessary, utilizing metrics such as Inception Score and Fréchet Inception Distance to assess performance accurately. The integration of MLFlow in our workflow allowed us to systematically manage and fine-tune hyperparameters, as demonstrated by the parallel coordinates plot that visualized the relationship between different parameters and model performance. We could also draw inspiration from the approach taken by the researchers behind LeafGAN by applying our Coral CycleGAN to additional, previously unseen coral datasets, to perform data augmentation. The follow-up test on the efficacy of classifiers for unhealthy corals could validate our GAN mode's performance.

Finally, deploying our models using Streamlit provided an interactive and user-friendly interface, enabling practical demonstration and accessibility. Future work will focus on cloud deployment to enhance scalability and reach a broader audience. By continuing to refine these models and expanding our dataset, we aim to improve their accuracy and effectiveness in monitoring and conserving coral reefs.

Reference

1. The CoralNet Dataset. (2019). Retrieved from https://coralnet.ucsd.edu/pages/help/api/
2. The FathomNet Dataset. (2023). Retrieved from https://fathomnet.org/fathomnet/#/explorer
3. The Underwater Bounding Box Dataset. (2021). Retrieved from https://www.kaggle.com/datasets/akshatsng/underwater-dataset-for-8-classes-with-label?select=valid
4. The Coral Health Dataset (2021). Retrived from https://www.kaggle.com/datasets/vencerlanz09/healthy-and-bleached-corals-image-classification
5. Mahmood, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Kendrick, G., & Fisher, R. B. (2018). Coral Reefs. In *Marine Ecosystems: Human Impacts on Biodiversity, Functioning and Services* (pp. 445–467). Elsevier. https://www.sciencedirect.com/science/article/pii/B9780128113189000211
6. Chen, Q., Beijbom, O., Chan, S., Bouwmeester, J., & Kriegman, D. (2021). A New Deep Learning Engine for CoralNet. In *Proceedings of the ICCV 2021 Workshop on Computer*

Vision for Ocean Vision (pp. 2983–2992). CVF Open Access. https://openaccess.thecvf.com/content/ICCV2021W/OceanVision/papers/Chen_A_New_Deep_Learning_Engine_for_CoralNet_ICCVW_2021_paper.pdf

7. Zhang, H., Gruen, A., & Li, M. (2022). Advanced Image Processing Techniques for Deep Learning in Underwater Coral Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2022, 343–350. https://isprs-annals.copernicus.org/articles/V-2-2022/343/2022/isprs-annals-V-2-2022-343-2022.pdf
8. Cap, Q. H., Uga, H., Kagiwada, S., & Iyatomi, H. (2020, February 24). LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis. *arXiv*. https://arxiv.org/abs/2002.10100